

EL580803634

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Resynchronizing Globally Unsynchronized Multimedia
Streams**

Inventor(s):
Anoop Gupta
Nosakhare D. Omoigui

ATTORNEY'S DOCKET NO. MS1-278US

TECHNICAL FIELD

This invention relates to networked client/server systems and to methods of streaming and rendering multimedia content in such systems. More particularly, the invention relates to systems and methods of resynchronizing globally unsynchronized media streams.

BACKGROUND OF THE INVENTION

Multimedia streaming—the continuous delivery of synchronized media data like video, audio, text, and animation—is a critical link in the digital multimedia revolution. Today, streamed media is primarily about video and audio, but a richer, broader digital media era is emerging with a profound and growing impact on the Internet and digital broadcasting.

Locally synchronized media (or just "synchronized media") means multiple media objects that share a common timeline. Video and audio are examples of synchronized media—each is a separate data stream with its own data structure, but the two data streams are played back in synchronization with each other. Virtually any media type can have a timeline. For example, an image object can change like an animated .gif file, text can change and move, and animation and digital effects can happen over time. This concept of synchronizing multiple media types is gaining greater meaning and currency with the emergence of more sophisticated media composition frameworks implied by MPEG-4, Dynamic HTML, and other media playback environments.

The term "streaming" is used to indicate that the data representing the various media types is provided over a network to a client computer on a real-time, as-needed basis, rather than being pre-delivered in its entirety before

1 playback. Thus, the client computer renders streaming data as it is received from a
2 network server, rather than waiting for an entire "file" to be delivered.

3 The same data can be simultaneously streamed to multiple client
4 computers. In some situations, it is very important that this data be rendered by
5 each of the multiple client computers at as close to the same time as possible
6 (referred to as the data streams being "globally synchronized"). For example, live
7 financial broadcasts (e.g., stock quotes or other reports) should be made available
8 to all recipients at as close to the same time as possible. If one computer is
9 rendering the streamed data before another, each of the data streams being
10 streamed to the two computers can be referred to as being "out of global
11 synchronization" or "globally unsynchronized".

12 Rendering of data streams can become globally unsynchronized for a
13 variety of different reasons. Bugs or other problems may exist in certain rendering
14 devices (e.g., audio or video cards and/or their corresponding drivers) that cause
15 the data on one computer to be rendered either slower or faster than expected. An
16 example is clock-rate differences in the audio cards of rendering computers. This
17 results in the data being rendered slower on one computer than on another, and the
18 streams becoming more and more out of synchronization as rendering continues.
19 Additionally, many modern computer systems are designed for the CPU to execute
20 multiple different programs or processes concurrently (e.g., multi-tasking). In
21 such a system, the CPU may spend a significant amount of time executing another
22 process at the computer, causing the execution of the instructions for rendering the
23 data stream at that computer to lag behind the execution of the instructions for
24 rendering the data stream at another computer.

1 The invention described below addresses these disadvantages of data
2 streaming, providing a way to resynchronize globally unsynchronized streams.

3 4 **SUMMARY OF THE INVENTION**

5 In a networked client/server system, the invention detects when media
6 being streamed from a server(s) to multiple clients becomes globally
7 unsynchronized. The invention then provides for resynchronization of the
8 globally unsynchronized streams.

9 According to one aspect of the invention, the invention uses timeline
10 alteration to resynchronize a media stream from a server computer to a client
11 computer in a network system. The invention monitors the media stream and
12 detects when the stream has become globally unsynchronized with a
13 corresponding media stream being streamed to another client computer(s). When
14 the stream has become globally unsynchronized, the invention alters the
15 presentation timeline for the stream so that it is resynchronized with the other
16 stream(s).

17 According to another aspect of the invention, the invention monitors how
18 much data (in terms of presentation time) is buffered at a client computer for a
19 data stream. If greater than a threshold amount of data is buffered, then the stream
20 is deemed to be globally unsynchronized with corresponding data streams to other
21 client computers.

22 According to another aspect of the invention, the timeline alteration occurs
23 at the client computer. In accordance with this aspect, the client computer alters
24 the timeline of a media stream(s) as necessary (either compressing or expanding
25 the timeline) to resynchronize the media streams.

1 According to another aspect of the invention, the timeline alteration occurs
2 at the server computer. In accordance with this aspect, either the client computer
3 or the server computer monitors the media streams to determine when the streams
4 have become globally unsynchronized. Once the streams are globally
5 unsynchronized, the server computer selects a different one of multiple media
6 streams corresponding to the media content being streamed to one of the client
7 computers. This different media stream has a different timeline (either
8 compressed or expanded) than the media stream that was previously being
9 streamed to that client computer. This new media stream is then streamed to the
10 client computer by the server computer in order to resynchronize the media
11 streams.

12 Additionally, the timeline alteration and/or the new stream selection can be
13 performed seamlessly, resulting in a much improved user experience – there are no
14 perceived pauses or discontinuities of the multimedia streams.

15 **BRIEF DESCRIPTION OF THE DRAWINGS**

16
17 Fig. 1 is a block diagram of a networked client/server system in accordance
18 with one embodiment of the invention.

19 Fig. 2 is a block diagram of a networked computer that can be used to
20 implement either a server or a client in accordance with one embodiment of the
21 invention.

22 Fig. 3 is a block diagram illustrating one embodiment of the invention, in
23 which resynchronization is performed by a client.

24 Fig. 4 illustrates methodological aspects of the invention according to an
25 embodiment implemented in a client computer system.

1 Fig. 5 is a block diagram illustrating one embodiment of the invention, in
2 which resynchronization is performed by a server.

3 Fig. 6 illustrates methodological aspects of the invention according to an
4 embodiment implemented in a server computer system.

5 6 **DETAILED DESCRIPTION**

7 **General Network Structure**

8 Fig. 1 shows a client/server network system and environment in accordance
9 with one embodiment of the invention. Generally, the system includes a network
10 server computer 10 and a plurality of network client computers 11. The computers
11 communicate with each other over a data communications network. The
12 communications network in Fig. 1 comprises a public network 12 such as the
13 Internet. The data communications network might also include local-area
14 networks and private wide-area networks.

15 Server computer 10 has access to streaming media content in the form of
16 different composite media streams. Some composite media streams might be
17 stored as files in a database or other file storage system 13. Other composite
18 media streams 14 might be supplied to the server on a "live" basis from other data
19 source components through dedicated communications channels or through the
20 Internet itself.

21 The composite media streams can be provided to the network client
22 computers 11 from the server computer 10. Alternatively, the network system may
23 include multiple server computers 10, with different composite media streams
24 being available from different server computers 10. Additionally, if the network
25

1 system includes multiple server computers 10, different media streams of a
2 composite media stream may be stored at different server computers 10.

3 4 **Streaming Media**

5 In this discussion, the term “composite media stream” describes
6 synchronized streaming data that represents a segment of multimedia content. The
7 composite media stream has a timeline that establishes the speed at which the
8 content is rendered. The composite media stream can be rendered to produce a
9 plurality of different types of user-perceivable media, including synchronized
10 audio or sound, video graphics or motion pictures, animation, textual content,
11 command script sequences, or other media types that convey time-varying
12 information or content in a way that can be sensed and perceived by a human. A
13 composite media stream comprises a plurality of individual media streams
14 representing the multimedia content. Each of the individual media streams
15 corresponds to and represents a different media type and each of the media
16 streams can be rendered by a network client to produce a user-perceivable
17 presentation using a particular presentation medium. The individual media
18 streams have their own timelines, which are synchronized with each other so that
19 the media streams can be rendered simultaneously for a coordinated multimedia
20 presentation. The individual timelines define the timeline of the composite
21 stream.

22 There are various standards for streaming media content and composite
23 media streams. The “Advanced Streaming Format” (ASF) is an example of such a
24 standard, including both accepted versions of the standard and proposed standards
25 for future adoption. ASF specifies the way in which multimedia content is stored,

1 streamed, and presented by the tools, servers, and clients of various multimedia
2 vendors. ASF provides benefits such as local and network playback, extensible
3 media types, component download, scalable media types, prioritization of streams,
4 multiple language support, environment independence, rich inter-stream
5 relationships, and expandability. Further details about ASF are available from
6 Microsoft Corporation of Redmond, Washington.

7 Regardless of the streaming format used, an individual data stream contains
8 a sequence of digital data sets or units that are rendered individually, in sequence,
9 to produce an image, sound, or some other stimuli that is perceived by a human to
10 be continuously varying. For example, an audio data stream comprises a sequence
11 of sample values that are converted to a pitch and volume to produce continuously
12 varying sound. A video data stream comprises a sequence of digitally-specified
13 graphics frames that are rendered in sequence to produce a moving picture.

14 Typically, the individual data units of a composite media stream are
15 interleaved in a single sequence of data packets. Various types of data
16 compression might be used within a particular data format to reduce
17 communications bandwidth requirements.

18 The sequential data units (such as audio sample values or video frames) are
19 associated with both delivery times and presentation times, relative to an arbitrary
20 start time. The delivery time of a data unit indicates when the data unit should be
21 delivered to a rendering client. The presentation time indicates when the value
22 should be actually rendered. Normally, the delivery time of a data unit precedes
23 its presentation time.

24 The presentation times determine the actual speed of playback. For data
25 streams representing actual events or performances, the presentation times

1 correspond to the relative times at which the data samples were actually recorded.
2 The presentation times of the various different individual data streams are
3 consistent with each other so that the streams remain coordinated and
4 synchronized during playback.
5

6 **Exemplary Computer Environment**

7 In the discussion below, the invention will be described in the general
8 context of computer-executable instructions, such as program modules, being
9 executed by one or more conventional personal computers. Generally, program
10 modules include routines, programs, objects, components, data structures, etc. that
11 perform particular tasks or implement particular abstract data types. Moreover,
12 those skilled in the art will appreciate that the invention may be practiced with
13 other computer system configurations, including hand-held devices,
14 multiprocessor systems, microprocessor-based or programmable consumer
15 electronics, network PCs, minicomputers, mainframe computers, and the like. In a
16 distributed computer environment, program modules may be located in both local
17 and remote memory storage devices.

18 Fig. 2 shows a general example of a computer 20 that can be used as a
19 network node or host in accordance with the invention. Computer 20 is shown as
20 an example of a computer that can perform the functions of either server computer
21 10 or a client computer 11.

22 Computer 20 includes one or more processors or processing units 21, a
23 system memory 22, and a system bus 23 that couples various system components
24 including the system memory 22 to processors 21.
25

1 The system bus 23 may be any of several types of bus structures including a
2 memory bus or memory controller, a peripheral bus, and a local bus using any of a
3 variety of bus architectures. The system memory includes read only memory
4 (ROM) 24 and random access memory (RAM) 25. A basic input/output system
5 (BIOS) 26, containing the basic routines that help to transfer information between
6 elements within computer 20, such as during start-up, is stored in ROM 24.
7 Computer 20 further includes a hard disk drive 27 for reading from and writing to
8 a hard disk, not shown, a magnetic disk drive 28 for reading from and writing to a
9 removable magnetic disk 29, and an optical disk drive 30 for reading from or
10 writing to a removable optical disk 31 such as a CD ROM or other optical media.
11 The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are
12 connected to the system bus 23 by an SCSI interface 32 or some other appropriate
13 interface. The drives and their associated computer-readable media provide
14 nonvolatile storage of computer readable instructions, data structures, program
15 modules and other data for computer 20. Although the exemplary environment
16 described herein employs a hard disk, a removable magnetic disk 29 and a
17 removable optical disk 31, it should be appreciated by those skilled in the art that
18 other types of computer readable media which can store data that is accessible by a
19 computer, such as magnetic cassettes, flash memory cards, digital video disks,
20 random access memories (RAMs) read only memories (ROM), and the like, may
21 also be used in the exemplary operating environment.

22 A number of program modules may be stored on the hard disk, magnetic
23 disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35,
24 one or more application programs 36, other program modules 37, and program
25 data 38. A user may enter commands and information into computer 20 through

1 input devices such as keyboard 40 and pointing device 42. Other input devices
2 (not shown) may include a microphone, joystick, game pad, satellite dish, scanner,
3 or the like. These and other input devices are connected to the processing unit 21
4 through an interface 46 that is coupled to the system bus. A monitor 47 or other
5 type of display device is also connected to the system bus 23 via an interface, such
6 as a video adapter 48. In addition to the monitor, personal computers typically
7 include other peripheral output devices (not shown) such as speakers and printers.

8 Computer 20 operates in a networked environment using logical
9 connections to one or more remote computers, such as a remote computer 49. The
10 remote computer 49 may be another personal computer, a server, a router, a
11 network PC, a peer device or other common network node, and typically includes
12 many or all of the elements described above relative to computer 20, although only
13 a memory storage device 50 has been illustrated in Fig. 2. The logical connections
14 depicted in Fig. 2 include a local area network (LAN) 51 and a wide area network
15 (WAN) 52. Such networking environments are commonplace in offices,
16 enterprise-wide computer networks, intranets, and the Internet. In the described
17 embodiment of the invention, remote computer 49 executes an Internet Web
18 browser program such as the "Internet Explorer" Web browser manufactured and
19 distributed by Microsoft Corporation of Redmond, Washington.

20 When used in a LAN networking environment, computer 20 is connected to
21 the local network 51 through a network interface or adapter 53. When used in a
22 WAN networking environment, computer 20 typically includes a modem 54 or
23 other means for establishing communications over the wide area network 52, such
24 as the Internet. The modem 54, which may be internal or external, is connected to
25 the system bus 23 via a serial port interface 33. In a networked environment,

1 program modules depicted relative to the personal computer 20, or portions
2 thereof, may be stored in the remote memory storage device. It will be
3 appreciated that the network connections shown are exemplary and other means of
4 establishing a communications link between the computers may be used.

5 Generally, the data processors of computer 20 are programmed by means of
6 instructions stored at different times in the various computer-readable storage
7 media of the computer. Programs and operating systems are typically distributed,
8 for example, on floppy disks or CD-ROMs. From there, they are installed or
9 loaded into the secondary memory of a computer. At execution, they are loaded at
10 least partially into the computer's primary electronic memory. The invention
11 described herein includes these and other various types of computer-readable
12 storage media when such media contain instructions or programs for implementing
13 the steps described below in conjunction with a microprocessor or other data
14 processor. The invention also includes the computer itself when programmed
15 according to the methods and techniques described below. Furthermore, certain
16 sub-components of the computer may be programmed to perform the functions
17 and steps described below. The invention includes such sub-components when
18 they are programmed as described. In addition, the invention described herein
19 includes data structures, described below, as embodied on various types of
20 memory media.

21 For purposes of illustration, programs and other executable program
22 components such as the operating system are illustrated herein as discrete blocks,
23 although it is recognized that such programs and components reside at various
24 times in different storage components of the computer, and are executed by the
25 data processor(s) of the computer.

Time-Scale Modification

As shown in Fig. 1, a network system in accordance with the invention includes a network server 10 from which one or more composite media streams are available. In the described embodiment of the invention, the media streams are stored as files on some type of data storage device accessible by the server.

The system also includes network clients 11. Generally, the network clients are responsive to user input to select or request identified composite media streams. In response to a request for a composite media stream, server 10 streams the requested composite media stream to the requesting network client in accordance with some known format such as ASF. The client renders the data streams to produce the multimedia content.

Additionally, a network client may also accept a speed designation or playback speed from a human user. The speed designation might be a speed factor relative to the original or default playback speed of the selected multimedia stream. For example, a speed factor of 1.2 indicates that the composite media stream is to be rendered at 1.2 times its original or default speed, thereby achieving time compression. A speed factor of 0.8 indicates that the composite media stream is to be rendered at 0.8 times its original or default speed, thereby achieving time expansion.

In response to the speed or playback designation from the user, the client requests a composite media stream from the server, indicating the desired type, level, or degree of timeline alteration. A plurality of composite streams are available at the server, corresponding to different possible timeline alterations. In response to the request from the client, the server begins streaming a composite

1 stream that has already had its timeline altered in accordance with the speed
2 designation. Alternatively, the server may not include multiple composite streams
3 for the different possible timeline alterations, in which case the client performs the
4 timeline alteration (although it should be noted that the server would then need to
5 speed up or slow down as appropriate its delivery of the media stream to the client
6 when the client alters the timeline).

7 With some types of media, such as video streams, timeline alteration
8 involves either omitting selected frames or modifying the presentation times of the
9 individual data units or video frames. In other cases, such as with audio streams,
10 the time-modification is more difficult—simply changing the presentation times
11 would alter the pitch of the original audio and make it unintelligible. Accordingly,
12 some type of audio processing technique is used to time-compress or time-expand
13 audio streams, while maintaining the original pitch of the audio—thereby
14 maintaining the intelligibility of the audio.

15 There are various known methods of audio time modification, commonly
16 referred to as “time-scale-modification,” most of which concentrate on removing
17 redundant information from the speech signal. In a method referred to as
18 sampling, short segments are dropped from the speech signal at regular intervals.
19 Cross fading or smoothing between adjacent segments improves the resulting
20 sound quality.

21 Another method, referred to as synchronized overlap add method (SOLA or
22 OLA), consists of shifting the beginning of a new speech segment over the end of
23 the preceding segment to find the point of highest cross-correlation (i.e., maximum
24 similarity). The overlapping frames are averaged, or smoothed together, as in the
25 sampling method.

1 Sampling with dichotic presentation is a variant of the sampling method
2 that takes advantage of the auditory system's ability to integrate information from
3 both ears. It improves on the sampling method by playing the standard sampled
4 signal to one ear and the "discarded" material to the other ear. Intelligibility and
5 compression increase under this dichotic presentation condition when compared
6 with standard presentation techniques.

7 More information regarding audio time modification is given in an article
8 that appeared in the March, 1997, issue of "ACM Transactions on Computer-
9 Human Interaction" (Volume 4, Number 1, pages 3-38) (1997). For purposes of
10 this disclosure, it can be assumed that audio time modification involves some
11 combination of changing individual data stream samples, dropping certain
12 samples, and adjusting presentation times of any samples that are actually
13 rendered.

14 The methods mentioned above are considered "linear" because all portions
15 of the speech signal are compressed or expanded uniformly. That is, timeline
16 alteration is applied consistently in time.

17 In addition, or alternatively, the speed designation might indicate one of a
18 plurality of different types or levels of non-linear timeline compression. As an
19 example, such levels might include a type of non-linear timeline compression that
20 removes audio pauses from a spoken presentation or periods of relative inactivity
21 from a video presentation. Another level, perhaps referred to as "summary"
22 compression, might retain only short periods of a presentation that are marked by
23 high energy levels in either audio or video portions of the content. With non-
24 linear techniques some segments of an individual or composite stream might be
25 more highly expanded or compressed (in time) than other segments. Furthermore,

1 the respective individual streams of a composite stream might be expanded or
2 compressed by different amounts relative to each other.

3 4 **Client-Based Resynchronization**

5 Fig. 3 illustrates an embodiment of the invention in which
6 resynchronization is performed by network client 11. Network server 10 streams a
7 composite media stream 101 to network client 11 (although not shown in Fig. 3,
8 other communications also take place bi-directionally between server 10 and client
9 11, such as control-oriented communications). The composite media stream has a
10 plurality of individual media streams as described above. For purposes of
11 discussion, it is assumed in this example that the composite media stream has an
12 audio stream and a video stream.

13 Each media stream has a timeline, and the timelines of the individual
14 streams are synchronized with each other so that the streams can be rendered in
15 combination to produce coordinated multimedia content at the network client 11.
16 The original timelines correspond to the original recording or rendition of the
17 multimedia material, so that rendering the streams according to their timelines
18 results in presentation speeds that closely match the speed of the original event or
19 performance. In the case of audio streams, the timelines preserve the original
20 speed and pitch of the original audio content.

21 The client computer has a demultiplexer component 104 that receives the
22 composite media stream and that separates out the individual media streams from
23 the composite format in which the data is streamed (such as ASF). The separate
24 media streams are temporarily buffered in buffers 102 and 103, from which an
25 audio stream 105 and a video media stream 106 are provided, respectively. The

1 individual media streams are sent to and received by respective decoders 108 and
2 109 that perform in accordance with the particular data format being employed.
3 For example, the decoders might perform data decompression.

4 The decoded data streams are then sent to and received by time
5 modification components: an audio timeline modification component 110 and a
6 video timeline modification component 111. These components receive input
7 from a human operator in the form of a speed designation as described above. The
8 timeline modification components change the timelines of the received media
9 streams in accordance with the user speed designation to achieve either time
10 compression or time expansion. With some types of media, such as video streams,
11 this involves either omitting selected frames or modifying the presentation times
12 of the individual data units or video frames. In other cases, such as with audio
13 streams, some type of audio processing technique as the SOLA technique
14 described above is used to time-compress or time-expand audio streams, while
15 maintaining the original pitch of the audio and to also retain the intelligibility of
16 the audio.

17 The timeline modification components 110 and 111 produce individual
18 media streams that are provided to and received by respective renderers 114 and
19 115. The rendering components render the streams in accordance with their
20 modified timelines, as the streams continue to be streamed from the network
21 server. In alternative embodiments of the invention, timeline modification
22 components 110 and 111 might be eliminated and their functions performed by
23 decoders 108 and 109.

24 The demultiplexer component 104 also has a synchronizer 107, coupled to
25 the timeline modification components 110 and 111 and the renderers 114 and 115

as illustrated, that monitors the media streams to ensure that they remain synchronized with the corresponding streams being delivered to other client computers (that is, remain globally synchronized). When synchronizer 107 detects that a media stream(s) being rendered at client 11 is out of synchronization with a corresponding media stream(s) being rendered at another client computer, the time modification performed by the timeline modification component 110 or 111 corresponding to at least one of the streams is altered to resynchronize the stream(s) with its counterpart(s) being delivered to the other client computer.

Note that the speed designation, provided by the user and possibly modified by synchronizer 107, dictates the rate at which the network client consumes the composite data stream. Because of this, the client communicates the speed designation to the network server when requesting a particular composite media stream. The server responds by streaming the composite media stream at a rate that depends on or is proportional to the speed designation provided by the user as modified by synchronizer 107. For example, for a speed factor of 2.0, the client consumes data at twice the normal rate. Accordingly, the server streams the composite media stream at twice its normal rate to meet the demands of the client.

In the described embodiment, the user and the synchronizer 107 are allowed to change the speed designation during rendering of the composite media stream. In some cases, however, it may not be possible to change the playback speed without interrupting the playback momentarily. If this is the case, playback resumes as soon as possible, beginning at a point that shortly precedes the point at which playback was discontinued. Thus, there is some overlap in the presentation—when the presentation resumes, the overlap provides context for the new content that follows.

1 Fig. 4 illustrates methodological aspects of the invention according to an
2 embodiment implemented in a client computer system.

3 A step 132 comprises monitoring the global synchronization of the media
4 streams. According to one implementation, this monitoring comprises the
5 synchronizer 107 monitoring how much presentation time (e.g., in seconds or
6 milliseconds) of data is buffered (e.g., in buffers 102 and 103) waiting to be
7 decoded and rendered. Alternatively, the monitoring may comprise the
8 synchronizer 107 monitoring the current presentation times of the devices
9 rendering the media streams (e.g., renderers 114 and 115 of Fig. 3). These current
10 presentation times are available from the renderers 114 and 115.

11 A step 134 comprises the synchronizer 107 checking whether the media
12 streams are out of global synchronization with corresponding media streams being
13 delivered to another client computer beyond a particular threshold value.
14 According to one implementation, this checking comprises determining the time
15 required to present all of the data that is stored in buffers 102 and 103 and
16 checking whether the required time exceeds a threshold value (e.g., 5 seconds
17 worth of data). The monitoring step 132 and checking step 134 are repeated until
18 the media stream becomes globally unsynchronized beyond the threshold value.
19 Alternatively, control information may be received from server 10 of Fig. 1 or
20 some other computer notifying client 11 what the current presentation times of the
21 devices rendering the media streams should be. Synchronizer 107 then determines
22 whether the difference between the actual presentation times and what the
23 presentation times should be exceeds the threshold value.

24 A step 136 comprises the synchronizer 107 determining an effective time
25 modification factor. The time modification factor identifies the alteration that will

1 be made to the timeline of a globally unsynchronized media stream to
2 resynchronize the media stream. The time modification factor can be determined
3 in any of a wide variety of manners. The time modification factor could be a
4 specific value (e.g., 20 milliseconds per second), or a relative value (e.g., 10
5 percent). The actual value (whether specific or relative) can vary, balancing the
6 intelligibility of the stream received by the user against the desire for quick
7 resynchronization. It is to be appreciated that the time modification factor
8 determined in step 136 can be a positive value (e.g., to speed up the presentation
9 of the media stream) or a negative value (e.g., to slow down the presentation of the
10 media stream).

11 A step 138 comprises altering the presentation timeline of one of the media
12 streams according to the time modification factor determined in step 136. In the
13 illustrated embodiment of Fig. 3, the synchronizer 107 modifies the time
14 modification factor used by the respective one of time modification components
15 110 and 111 by adding the modification factor determined in step 136 to the speed
16 factor currently being used for the media stream. Synchronizer 107
17 resynchronizes the media streams by using the time modification factor to slow
18 down (or speed up) the rendering of the media streams by renderers 114 and 115.

19 A step 140 comprises monitoring the global synchronization of the media
20 stream, while a step 142 comprises checking whether the media stream is back in
21 global synchronization (that is, whether the media stream has been globally
22 resynchronized). Given the modification of the presentation timeline in step 138,
23 eventually the media stream should be globally resynchronized.

24 According to one implementation, global resynchronization is determined
25 according to the same threshold value used to determine becoming globally

1 unsynchronized in step 134. For example, if the presentation times no longer
2 differ by greater than the threshold value used in step 134 (or the presentation time
3 of buffered data no longer exceeds the threshold value), then the data is
4 resynchronized. According to an alternate embodiment, this "resynchronized
5 threshold value" in step 142 is different than the "unsynchronized threshold value"
6 in step 134. By way of example, the threshold value in step 134 may be 5
7 seconds, whereas the resynchronization threshold value may be 3 seconds. By
8 making these resynchronized and unsynchronized threshold values different, the
9 system can avoid continually switching back and forth between being globally
10 synchronized and globally unsynchronized.

11 A step 144 comprises determining a new time modification factor once the
12 media stream is globally resynchronized. According to one implementation, this
13 new time modification factor is the difference between the current timeline speed
14 factor and the speed factor of the timeline prior to its alteration in step 138.
15 Alternatively, the modification factor in step 144 may be different to try to correct
16 the problem causing the stream to become globally unsynchronized. For example,
17 if the modification factor in step 136 was +10 milliseconds per second, the
18 modification factor in step 144 may be -8 milliseconds per second.

19 A step 146 comprises modifying the present timeline according to the new
20 modification factor. The process then returns to step 132, to continue monitoring
21 the global synchronization.

22 According to an alternate embodiment of the invention, steps 140 through
23 146 are optional and need not be included. In this embodiment, after the timeline
24 is modified in step 138, the process returns to step 132 to continue monitoring the
25 global synchronization.

1 Alternatively, other solutions which do not require timeline modification
2 may also be used to resynchronize the media streams. According to one such
3 alternative implementation, one of the streams can "jump ahead" to catch up to the
4 other stream. For example, if synchronizer 107 of Fig. 3 detects that the media
5 stream is being rendered ahead of a corresponding stream at other client systems
6 (e.g., out of global synchronization beyond the threshold amount in step 134), then
7 synchronizer 107 can have the stream "jump ahead" to a later presentation time.
8 For example, portions of the data (e.g., two or three seconds worth of data) can be
9 deleted from buffers 102 and 103. The amount of this "jump" can be a fixed
10 amount (e.g., two or three seconds), or alternatively a variable amount (e.g., 40%
11 of the data in the buffers 102 and 103). After such a jump, the process returns to
12 step 132 to continue to monitor the global synchronization and determine whether
13 the media streams are out of global synchronization.

14 According to another alternative implementation, one of the streams is
15 stopped temporarily. For example, if synchronizer 107 detects that the media
16 stream is being rendered behind a corresponding stream at other client systems
17 (e.g., out of global synchronization beyond the threshold amount in step 134), then
18 synchronizer 107 can temporarily pause the stream. For example, synchronizer
19 107 can communicate with buffers 102 and 103 to stop providing audio and video
20 data to decoders 108 and 109 for an amount of time (e.g., two or three seconds).
21 The duration of this pause caused by not providing data to decoders 108 and 109
22 can be a fixed amount (e.g., two or three seconds), or alternatively a variable
23 amount (e.g., 30% of the time that would be required to render all of the data
24 currently in buffers 102 and 103). After such a pause, the process returns to step
25

1 132 to continue to monitor the global synchronization and determine whether the
2 media streams are out of global synchronization.

3 4 **Server-Based Resynchronization**

5 In various embodiments of the invention, the global resynchronizing of a
6 media stream(s) can be performed in the client as described above, in the server, or
7 in both the client and server. In the network environment, it is often desirable to
8 avoid performing any significant timeline modification in the server. Otherwise,
9 the server could quickly become overloaded with requests from multiple clients.

10 However, in some cases it may be desirable to store multiple versions of
11 media streams at a server and to select particular versions of the media streams
12 depending on the timeline requirements of the client (based both on requirements
13 designated by the user and resynchronization requirements). One advantage of
14 this method is that it can require comparatively less communications bandwidth
15 between the server and client.

16 As a general example, a server might store a plurality of media streams that
17 are live modifications of a "live" stream having timelines modified by different
18 factors. When a client requests a composite media stream, the server selects the
19 version of the media stream whose timeline most closely accords with the speed
20 designation set by the user. If the timeline does not exactly match the speed
21 designation, the client can perform further timeline modification.

22 Fig. 5 illustrates a more specific example. In this embodiment, a server 300
23 stores multiple media streams 301 corresponding to specific multimedia content
24 302. The media streams are of different types, such as audio and video. In Fig. 5,
25 audio streams are designated by the letter "A" and video streams are designated by

1 the letter "V". Any combination of a single audio stream and a single video
2 stream can be rendered to produce the multimedia content.

3 The various individual data streams have timelines that are modified by
4 different degrees. The speed factors are indicated in Fig. 5. In this embodiment,
5 the audio and corresponding video streams are organized as pairs, each pair
6 forming a composite media stream having a timeline that has been modified by a
7 factor of 0.5, 1.0, or 1.5.

8 When a client 303 requests multimedia content from server 300, the client
9 identifies both the content and the speed factor. In response, a selector component
10 304 of the server selects the audio and video streams that have timelines most
11 closely approximating the identified speed factor, and combines those individual
12 media streams to form the composite media stream. The resulting composite
13 media stream is then sent to the client. When the timeline is accelerated, this
14 saves bandwidth in comparison to sending an unaltered composite media stream
15 having a higher streaming rate to meet the accelerated consumption demands of
16 the client.

17 When the user changes the playback speed or a media stream needs to be
18 globally resynchronized, the client requests a new media stream that most closely
19 corresponds to the requested or resynchronization speed. Playback is resumed in
20 the new stream at the same point (relative to the subject content) at which it was
21 discontinued in the old stream. Thus, the new stream is initiated at some
22 intermediate point rather than at the beginning. When the streams are linearly
23 altered, it is not difficult to determine the appropriate presentation time in the new
24 stream. Specifically, the point in the new timeline equals
25 $\text{oldtime}(\text{oldfactor}/\text{newfactor})$, where oldtime is the presentation time in the first

1 media stream at which the speed change is to occur, oldfactor is the playback
2 speed or factor of the old media stream, and newfactor is the playback speed or
3 factor of the new media stream.

4 Fig. 6 illustrates methodological aspects of the invention according to an
5 embodiment implemented in a server computer system.

6 A step 160 comprises the server system receiving an "out of global
7 synchronization" indication from a client system. In this embodiment, the client
8 system monitors the global synchronization of the media streams and determines
9 when the streams are globally unsynchronized, analogous to steps 132 and 134 of
10 Fig. 4. When the stream becomes globally unsynchronized, an "out of global
11 synchronization" indication (e.g., a message) is sent to the server system.
12 Alternatively, synchronization information (e.g., the current presentation times of
13 the media streams) could be provided to the server system and the server system
14 could determine when the media stream becomes globally unsynchronized.

15 A step 162 comprises determining an effective time modification factor.
16 This time modification factor is determined analogous to step 136 above.

17 A step 164 comprises selecting a new version of one of the media streams
18 based on the effective time modification factor determined in step 162. The new
19 version could be of either the slower or the faster (or both) stream, analogous to
20 step 138 above. One of the multiple versions stored at the server is selected for
21 presentation. According to one implementation, the effective time modification
22 factor is compared to the time modification factors of each of the multiple streams
23 stored at the server and the version of the media stream having the factor closest to
24 the determined effective time modification factor is selected for delivery to the
25 client system.

1 A step 166 comprises checking whether a "globally resynchronized"
2 indication is received. The resynchronized indication (e.g., a message) is sent by
3 the client system when the media stream has been globally resynchronized.
4 Alternatively, synchronization information (e.g., the current presentation times of
5 each of the media streams) could be provided to the server and the server could
6 determine when the media stream has been globally resynchronized.

7 A step 168 comprises selecting a new version of the media stream to be
8 presented to the client system. According to one implementation, the new version
9 selected in step 168 is the same version as was being provided prior to the new
10 selection in step 164.

11 The process then ends, until another "out of global synchronization"
12 indication is received.

13 According to an alternate embodiment of the invention, steps 166 and 168
14 are optional and need not be included. In this embodiment, after the new version
15 of the media stream is selected in step 164, the process ends, until another "out of
16 global synchronization" indication is received.

17 Alternatively, other solutions which do not require timeline modification
18 may also be used to resynchronize the media streams. For example, server 300 of
19 Fig. 5 may "jump ahead" in a stream or pause a stream, analogous to the
20 discussions above regarding client 11 of Fig. 3, to globally resynchronize the
21 streams.

22 In the discussions above, regarding both client-based and server-based
23 resynchronization, the invention is described as working with a composite media
24 stream. It should be noted that alternate embodiments of the invention can be used
25

1 with a single media stream being provided to a client rather than a composite of
2 multiple media streams.

3 Also in the discussions above, regarding both client-based and server-based
4 resynchronization, a single time modification factor to be used to resynchronize
5 the globally unsynchronized media stream. Alternatively, multiple different time
6 modifications factors may be used to resynchronize the stream, such as by starting
7 with a more aggressive factor (e.g., a larger value) to more quickly resynchronize,
8 and then changing to a less aggressive factor (e.g., a smaller value) as the stream
9 gets closer to resynchronization.

10 Also in the discussions above, regarding both client-based and server-based
11 resynchronization, timeline modification is used to alter the timeline of one of the
12 media streams. Alternatively, multiple different media streams can be altered
13 simultaneously. For example if a first media stream to a first client lags behind
14 and becomes out of global synchronization with a second media stream to a
15 second client, then timeline alteration could be used to both speed up the first
16 media stream and slow down the second media stream.

17 Conclusion

18 The invention described above provides resynchronization of globally
19 unsynchronized media streams. The described techniques monitor the
20 synchronization of media streams being streamed to different computers and
21 detect when the streams become globally unsynchronized, and advantageously
22 resynchronizes the streams. In one implementation, the presentation timeline of at
23 least one of the streams is then modified to either increase or decrease the speed of
24 the rendering of the media stream, as appropriate. This timeline modification
25

1 allows the stream to become resynchronized in a manner that maintains a high-
2 quality user experience.

3 Although the invention has been described in language specific to structural
4 features and/or methodological steps, it is to be understood that the invention
5 defined in the appended claims is not necessarily limited to the specific features or
6 steps described. Rather, the specific features and steps are disclosed as preferred
7 forms of implementing the claimed invention.

MSI-278US.PAT.APP.DOC